Alexandria Engineering Journal (2023) 77, 95–107

HOSTED BY

Alexandria University

Alexandria Engineering Journal

www.elsevier.com/locate/aej
www.sciencedirect.com

ORIGINAL ARTICLE

5G in healthcare: Matching game-empowered intelligent medical network slicing

Chenjing Tian [a], Haotong Cao [b], Sahil Garg [c], Georges Kaddoum [c], Mohammad Mehedi Hassan [d], Jun Xie [a,*]

[a] *Institute of Command and Control Engineering, Army Engineering University, Nanjing, Jiangsu, China*
[b] *Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu, China*
[c] *Electrical Engineering Department, École de Technologie Supérieure, Montreal, Canada*
[d] *Department of Information Systems, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia*

Received 4 March 2023; revised 24 May 2023; accepted 11 June 2023
Available online 3 July 2023

KEYWORDS

Medical network slicing (MNS);
5G healthcare;
Quality of service(QoS);
Virtual network embedding (VNE);
Matching theory

Abstract Improving the healthcare system is imperative for increasing efficiency and reducing costs. The rapid adoption of Internet of Medical Things (IoMT) has facilitated a broad range of healthcare applications and services, from real-time and critical care monitoring to telemedicine. These services typically have highly distinct quality of service (QoS) requirement for communication networks and thus cannot be served with a uniform network. In such a context, network slicing technologies in 5G can be employed to create virtually independent and customized communication networks for these use cases to meet their QoS requirement. In this wok, we model network slicing for healthcare services as a virtual network embedding (VNE) problem and propose a two-sided matching theory-based virtual network embedding (MT-VNE) solution. In MT-VNE, four novel preference indexes are devised to construct differential preference lists. The deferred acceptance and modified shortest path-based algorithms are utilized to perform the virtual node and link mapping, respectively. Extensive simulations demonstrate that MT-VNE outperforms other baselines in terms of accepting more healthcare services and effectively utilizing physical network resources. Moreover, MT-VNE also significantly reduces service embedding time.

1. Introduction

In recent years, the application of 5G technology into healthcare has given rise to numerous advanced healthcare use cases, such as telemedicine, medical image transmission, medical monitoring, remote surgery, etc. However, these use cases often have distinct or conflicting quality of service (QoS)

* Corresponding author.
E-mail addresses: tianchenjing17@nudt.edu.cn (C. Tian), haotong.cao@polyu.edu.hk (H. Cao), sahil.garg@ieee.org (S. Garg), georges.kaddoum@etsmtl.ca (G. Kaddoum), mmhassan@ksu.edu.sa (M.M. Hassan), xiejun@aeu.edu.cn (J. Xie).

Peer review under responsibility of Faculty of Engineering, Alexandria University.

https://doi.org/10.1016/j.aej.2023.06.041

requirements for the communication networks that support them. For example, remote surgery requires a support network with ultra-low latency and ultra-high reliability [1], whereas medical image transmission services require enhanced network broadband but are less sensitive to traffic delay [2]. In addition, medical monitoring services need to provide network connectivity for a large number of sporadically active Internet of Medical Things (IoMT) devices [3], such as wearable and implantable medical devices, as well as smart sensing remote and in–house monitoring devices.

The diverse QoS requirements of the medical services illustrate that the traditional "one size fits all" network provisioning approach is no longer adequate for the current healthcare industry. Therefore, in this context, 5G network slicing technologies can be utilized to facilitate the coexistence of multiple medical services with varying requirements on a shared underlying physical network. By providing separate virtual networks with specific QoS guarantees for each medical service, network slicing enables efficient resource allocation and better management of network resources, ensuring that each medical service can operate with the desired level of performance and security.

Fig. 1 illustrates the network slicing implemented for healthcare services. In this scenario, each healthcare service is abstracted as a logical virtual network (VN), where virtual nodes can be identified as communication instances in healthcare services such as hospitals, schools, and communities. These virtual nodes are interconnected through virtual links, thereby forming a virtual topology. Moreover, each VN is configured with specific resource requirements, such as processing, storage, and bandwidth resources, that correspond to the demand of the corresponding service. Based on network virtualization (NV) technology, the hardware resources of underlying 5G distributed cloud networks are virtualized into pools of computing/networking-related resources. This innovative approach enables the creation of multiple medical slices with varying characteristics that can be co–hosted on the same substrate network (SN). The virtual resources in physical networks are allocated to each slice to provide corresponding services, as illustrated in the "underlying substrate network" in Fig. 1. Therefore, the network slicing for healthcare services can be modeled as a virtual network embedding (VNE) problem.

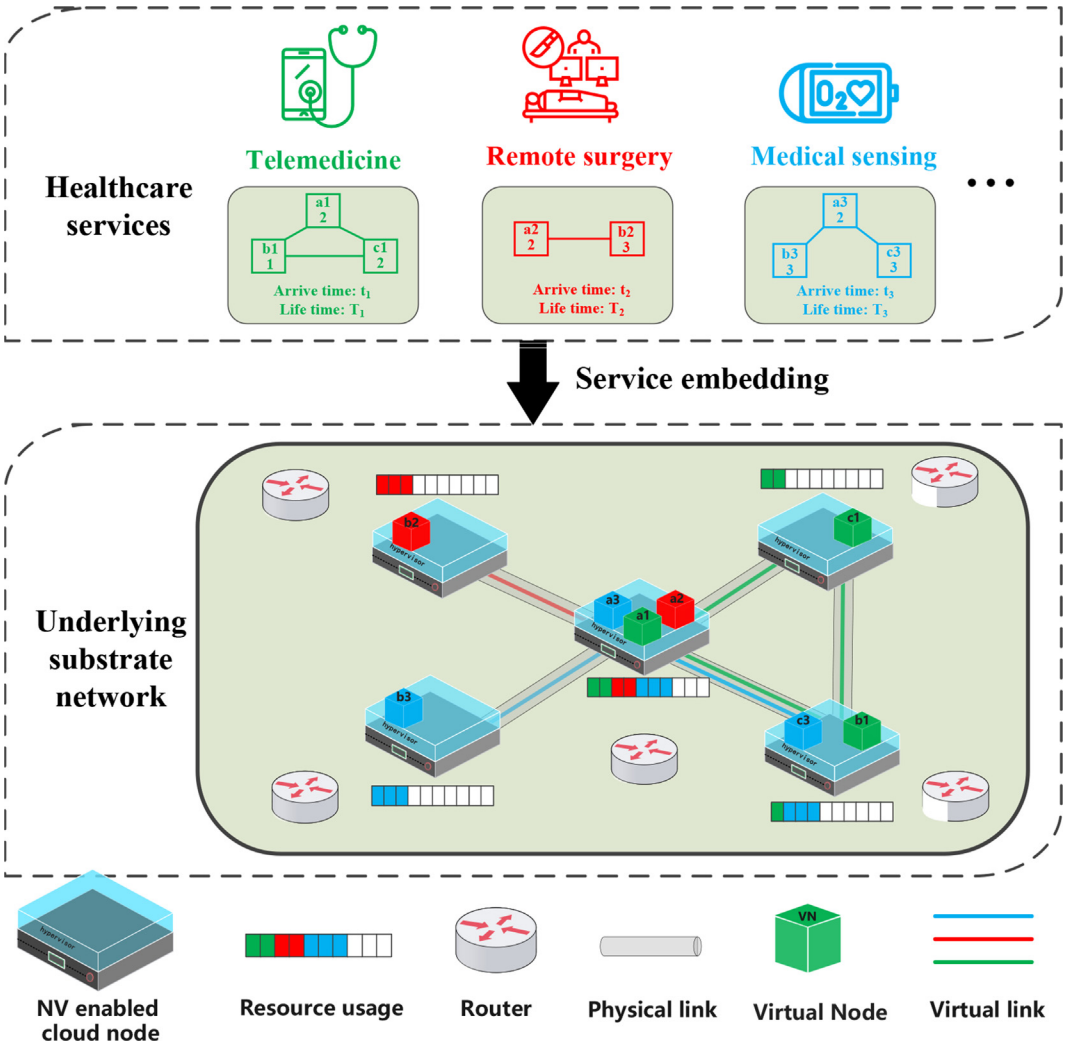

Fig. 1 Network slicing for healthcare services..

Traditional research on VNE provides insights into medical network slicing. By considering various objectives such as optimizing resource utilization efficiency, minimizing energy costs, or maximizing economic revenue, researchers have modeled VNE as diverse optimization problems, attempting to derive exact solutions with optimization solvers. However, the NP-hard nature of VNE problems makes it time-consuming to obtain exact VNE solutions. Consequently, researchers resort to heuristic and meta-heuristic methods to balance the trade-off between VNE result quality and computational complexity. Representative heuristic VNE algorithms include node ranking-based and subgraph isomorphism detection-based methods [4,5], while meta-heuristic algorithms involve particle swarm optimization (PSO)-based and genetic algorithm (GA)-based methods [6–8]. With the increasing application of deep reinforcement learning (DRL) in communication and networking [9], researchers have recently introduced DRL algorithms to solve VNE problems [10–12].

Apart from their similarities, healthcare network slicing-oriented VNE and traditional VNE problems also exhibit some distinct characteristics. For example, certain medical services, such as remote surgery, emergency rescue, and medical robots, often require low latency to ensure their proper functioning [13]. Therefore, it is necessary to consider the communication latency requirements between virtual nodes when performing healthcare network slicing. Furthermore, the virtual nodes in the logical medical network are commonly concentrated in densely populated regions, such as hospitals, communities, and schools, which typically possess static geographical positions. As a result, healthcare services may impose location constraints in VNE process.

Traditional methods have proven insufficient to address the challenges posed by latency requirements and location constraints in healthcare network slicing. As a result, this study proposes the matching game theory-based VNE (MT-VNE) algorithm, which integrates the two-sided matching theory into VNE. In MT-VNE, four preference indexes are devised to construct preference lists. Then, the obtained preference list information is used to execute virtual node mapping with deferred acceptance algorithm [14]. Finally, the modified shortest path-based algorithm performs the virtual link mapping. The main contributions of this work are summarized as follows.

1. This work aims to address the problem of service provisioning for healthcare use cases. To meet the diverse network requirements of different use cases, we incorporate network slicing architecture into the problem and model it as a virtual network embedding process. Additionally, we analyze the distinct characteristics of medical network slicing and take these characteristics into account during the problem-solving process.
2. Given the inefficiency of traditional VNE methods, we have introduced two-sided matching theory into medical network slicing and proposed the MT-VNE algorithm. In MT-VNE, we have devised four preference indexes to construct separate preference lists for each substrate/virtual node. The obtained preference lists provide more targeted preference relationships between virtual and substrate nodes, leading to more reasonable VNE results.
3. Extensive simulations were conducted to compare the performance of MT-VNE with several baselines [5,6,15]. The experimental results showed that MT-VNE outperformed the baselines in terms of service acceptance ratio and physical resources utilization, especially when taking into account the delay constraint. In addition, MT-VNE also demonstrated a significant reduction in VNE execution time. Besides, we also conducted a comparative experiment to illustrate the effect of the four preference indexes in MT-VNE.

The remainder of this paper is organized as follows. Section 2 discusses related works. Then, in Section 3, we present the healthcare network slicing problem formulation. Section 4 briefly reviews the two-sided matching theory, while Section 5 provides a detailed explanation of the MT-VNE algorithm. In Section 6, we present the simulation configurations and experimental results. Finally, in Section 7, we conclude this paper and suggest several promising research directions. Important notations of this paper are reported in Table 1.

2. Related works

2.1. Healthcare network slicing

The technology of network slicing comprises a pivotal ingredient of the 5G and forthcoming communication systems, facilitating the partition of the underlying physical network into several logical virtual networks that are tailored to distinct services. With the medical industry experiencing rapid advancements, a plethora of innovative medical services have

Table 1 Key VNE notations.

Notation	Description
$G^S = \{N^S, L^S\}$	Substrate network.
$G^V = \{N^V, L^V\}$	Virtual network.
$\mathrm{CPU}_{n^S}, \mathrm{Loc}_{n^S}$	Unoccupied CPU resource and node location of substrate node n^S.
$\mathrm{BW}_{l^S}, \mathrm{DL}_{l^S}$	Unoccupied bandwidth resource and propagation delay of substrate link l^S.
$\mathrm{CPU}_{n^V}, \mathrm{Loc}_{n^V}, \mathrm{LDC}_{n^V}$	CPU demand, preferred location, and location constraint of virtual node n^V.
$\mathrm{BW}_{l^V}, \mathrm{DLC}_{l^V}$	Bandwidth demand and maximum propagation delay tolerance of virtual link l^V.
$X_{n^S}^{n^V}$	Binary variable indicates if n^V is embedded onto n^S.
$Y_{p^S}^{l^V}$	Continuous variable specifies the amount of bandwidth that p^S allocated to l^V.
P_{S2V}, P_{V2S}	Preference lists of substrate nodes to virtual nodes, and virtual nodes to substrate nodes, respectively.
$Pval_{V2S}, Pval_{S2V}$	Preference value of virtual nodes to substrate nodes, and substrate nodes to virtual nodes, respectively.
R_{V2S}	Resource preference index.
D_{V2S}	Location preference index.
$D_{V2S}^{dir}, D_{V2S}^{adj}$	Direct and adjacent location preference index.
E_{S2V}	Resource utilization efficiency index.
L_{S2V}	Link connectivity index.

surfaced that are beyond the scope of conventional networks. Hence, scholars have delved into the realm of network slicing to grapple with these quandaries. Tebe et al. proposed an optimization method for maximizing the medical data throughput of different slices in a mobile hospital system in [16]. The work [17] constructed a fingerprinting based architecture to quickly customize resources for slices to meet the reliability requirements of smart health applications. In [18], Lin *et al.* investigated resource allocation in a multi-access edge computing-based cellular network to optimize the performance of mobile health applications. However, most of these works did not consider the QoS and location requirements specific to healthcare services. In this paper, we incorporate delay and location constraints into the system model to make it more practical and applicable to the healthcare services.

2.2. Virtual network embedding methods

Heuristic VNE algorithms. Due to the dynamic and unpredictable nature of virtual network request (VNR) arrivals in online VNE problems, the execution time plays a crucial role in achieving efficient and timely service provisioning. Hence, some works have focused on employing heuristic solutions such as node ranking and subgraph isomorphism detection based methods to solve VNE problems. Node ranking-based methods [19,5] divide the VNE procedure into two phases, namely node mapping and link mapping, and subgraph isomorphism detection based methods solve VNE by identifying an isomorphic subgraph within the SN that satisfies the VN requirements. Building on the node ranking VNE framework, Cao *et al.* [5] proposed a novel multiple network topology attributes and network resource-considered (NTANRC) algorithm. In this work, NTANRC adopted the form of Coulomb's law and Google PageRank algorithm to obtain stable node ranking lists.

Meta-heuristic VNE algorithms. Some researchers model VNE as a combinatorial optimization problem which seeks the optimal solution within a discrete solution space. However, finding the exact solution for such a problem with large VNE requests is a challenging task. To address this, meta-heuristics such as genetic algorithms, particle swarm optimization (PSO), or simulated annealing are often employed. These methods aim to obtain near-optimal solutions by iteratively improving a candidate solution based on a given quality measure. Wang *et al.* [6] constructively introduced the PSO into VNE (CPSO-VNE). CPSO-VNE encoded virtual node mapping and link mapping as particle positions. The velocity is defined as a matrix, with its elements indicating the probability of selection for each substrate node by virtual nodes. Moreover, this study employed a stepwise solution construction strategy to harmonize the mapping of virtual nodes and links in a single stage [6].

Deep reinforcement learning (DRL)-based VNE algorithms. By modeling the VNE problem as a Markov decision process, researchers [10,11,15] have explored the utilization of DRL methods for solving it. In DRL-based approaches, the knowledge of SN and VNR are represented as the input state of the agent. The agent then performs actions to allocate physical nodes, links, and related resources to the VNRs for service provisioning. In [15], Wang *et al.* proposed a temporal-difference learning-based VNE algorithm (TD-VNE) that aims to maximize the long-term revenue of service provider. TD-VNE leverages a node ranking-based VNE algorithm to generate multiple node-mapping candidates for a VNR, and selects the candidate with the highest "state-action value" as the virtual node embedding decision.

The above-mentioned VNE algorithms have demonstrated satisfactory performance in simulation experiments. However, they exhibit several shortcomings that render them inadequate for healthcare network slicing.

1. In node ranking-based algorithms, the ranking values of substrate/virtual nodes are calculated based on their respective topology and resource characteristics, resulting in all substrate/virtual nodes having identical preferences without expressing their differential inclination to the other parts.
2. The PSO-based algorithms own higher computational complexity compared with heuristic algorithms, which leads to a longer VNE execution time. However, the execution time is of great significance for online healthcare slicing problem since the VNRs' arrival time is dynamic and cannot be known beforehand [20].
3. The DRL-based algorithm is founded on deep neural networks, and the representation structure of input state can hardly be changed once the neural network is determined. Therefore, the trained DRL model is not universally applicable and needs to be redesigned and retrained when applied to different physical networks.

In this study, the proposed MT-VNE algorithm is designed to overcome these limitations and is suitable for healthcare network slicing. The approaches discussed in this section [5,6,15] will serve as the baselines to compare with the proposed MT-VNE algorithm.

3. System model and problem formulation

3.1. System model

The underlying SN of the system can be aptly represented as an undirected graph, referred to as $G^S = \{N^S, L^S\}$. Here, N^S comprises a collection of substrate nodes n^S that constitute the substrate network, while L^S comprises all communication links l^S. Each substrate node n^S is distinguished by its unoccupied CPU resources CPU_{n^S} Besides, the location Loc_{n^S} represents the geophysical position of the NV-enabled substrate node. The substrate links are characterized by their unoccupied bandwidth resource BW_{l^S} and link propagation delay DL_{l^S}. Moreover, all paths between different substrate nodes, devoid of any loops, are denoted by the set of P^S, with each specific path $p^S \in P^S$.

Similarly, the representation of medical services by the VN can also be modeled as a weighted undirected graph $G^V = \{N^V, L^V\}$. Here, N^V and L^V correspond to the set of virtual nodes n^V and virtual links l^V, respectively. The virtual node n^V possesses attributes such as CPU demand CPU_{n^V}, preferred location Loc_{n^V}, and location constraint LDC_{n^V}. Among these, Loc_{n^V} can denote the geographical position of virtual nodes (e.g., schools, apartment complexes, and other communities that may be involved in medical services), and LDC_{n^V} indicates the maximum distance that n^V can accept from the embedded substrate node. Furthermore, l^V is characterized

by its bandwidth demand BW_{l^V} and maximum propagation delay tolerance DLC_{l^V}.

3.2. Problem formulation

The VNE can be formulated as a mixed integer programming problem (MILP) that contains two variables $X_{n^S}^{n^V}$ and $Y_{p^S}^{l^V}$. The binary variable $X_{n^S}^{n^V}$ indicates whether the virtual node n^V is mapped onto the physical node n^S. If yes, $X_{n^S}^{n^V} = 1$; otherwise, $X_{n^S}^{n^V} = 0$. Variable $Y_{p^S}^{l^V}$ represents the amount of bandwidth that p^S allocated to l^V. The objective is to minimize the VNE cost while balancing the SN load [21]. The VNE process must also satisfy node constraints, aggregated bandwidth constraints, resource constraints, location constraints, propagation delay constraints, and domain constraints. Overall, the VNE problem can be formulated as:

Objective function:

$$\begin{aligned} min & \sum_{n^S \in N^S} \frac{\alpha}{\mathrm{CPU}_{n^S}+\delta} \sum_{n^V \in N^V} X_{n^S}^{n^V} \cdot \mathrm{CPU}_{n^V} \\ & + \sum_{l^S \in L^S} \frac{\beta}{\mathrm{BW}_{l^S}+\delta} \sum_{p^S \in P^S} \sum_{l^V \in L^V} Y_{p^S}^{l^V} \mathbb{I}_{\{l^S \in p^S\}} \end{aligned}, \tag{1}$$

where $\mathbb{I}_{\{\cdot\}}$ is an indicator function. It takes the value 1 if the conditions in parentheses are met and 0 otherwise. The weight coefficients α and β are tunable non-negative weights to balance the importance of CPU and bandwidth resources, respectively, with $\alpha + \beta = 1$. The constant δ is a small positive number that prevents the denominator from being zero. The objective function aims to minimize the cost of a single VN and balance the load of the SN.

Subject to:

Node Constraints:

$$\sum_{n^S \in N^S} X_{n^S}^{n^V} = 1, \forall n^V \in N^V. \tag{2}$$

$$\sum_{n^V \in N^V} X_{n^S}^{n^V} \leqslant 1, \forall n^S \in N^S. \tag{3}$$

Constraint (2) ensures that each virtual node is mapped onto a substrate node. Constraint (3) restricts that different virtual nodes in a single VNR can not embed onto the same substrate node.

Bandwidth Constraints:

$$\sum_{p^S \in P^S} Y_{p^S}^{l^V} = \mathrm{BW}_{l^V}, \forall l^V \in L^V. \tag{4}$$

$$\sum_{p^S \in P^S} Y_{p^S}^{l^V} (X_{n_1^{p^S}}^{n_1^{l^V}} X_{n_2^{p^S}}^{n_2^{l^V}} + X_{n_1^{p^S}}^{n_2^{l^V}} X_{n_2^{p^S}}^{n_1^{l^V}}) = \mathrm{BW}_{l^V}, \forall l^V \in L^V. \tag{5}$$

Eq. (4) states that the sum of the allocated bandwidth on each physical path must equal the bandwidth demand of the corresponding virtual link. In Eq. (5), $n_1^{p^S}$ and $n_2^{p^S}$ represent the end substrate nodes of p^S, while $n_1^{l^V}$ and $n_2^{l^V}$ are the end nodes of l^V. Therefore, constraint (5) ensures that the total bandwidth allocated to l^V from substrate paths connecting the mapping nodes of $n_1^{l^V}$ and $n_2^{l^V}$ equals l^V's bandwidth demand [22]. The combination of constraints (4) and (5) ensures that l^V's bandwidth demand can only be allocated to paths connecting substrate nodes where l^V's end virtual nodes are mapped to.

Resource Constraints:

$$\sum_{n^V \in N^V} X_{n^S}^{n^V} \cdot \mathrm{CPU}_{n^V} \leqslant \mathrm{CPU}_{n^S}, \forall n^S \in N^S. \tag{6}$$

$$\sum_{l^V \in L^V} \sum_{p^S \in P^S} Y_{p^S}^{l^V} \cdot \mathbb{I}_{\{l^S \in p^S\}} \leqslant \mathrm{BW}_{l^S}, \forall l^S \in L^S. \tag{7}$$

Constraints (6) and (7) ensures that the virtual nodes and links are mapped onto the substrate nodes and links with sufficient resources.

Location Constraint:

$$\sum_{n^S \in N^S} \mathrm{D}(\mathrm{Loc}_{n^V}, \mathrm{Loc}_{n^S}) \cdot X_{n^S}^{n^V} \leqslant \mathrm{LDC}_{n^V}, \forall n^V \in N^V. \tag{8}$$

Medical network slicing may require a location constraint to limit the distance between substrate nodes and virtual nodes. In (8), $\mathrm{D}(\cdot)$ is a function used to measure the Euclidean distance between n^V and n^S. The distance constraint ensures that the virtual nodes and their target substrate nodes satisfy the location constraint of virtual nodes.

Delay Constraint:

$$\sum_{l^S \in p^S} \mathrm{DL}_{l^S} \leqslant \mathrm{DLC}_{l^V}, \forall l^V \in L^V, \forall Y_{p^S}^{l^V} \neq 0. \tag{9}$$

Constraint (9) indicates that virtual link l^V can only be mapped onto the substrate paths in which the sum of physical links' propagation delay is less than or equal to the virtual link's propagation delay constraint.

Domain Constraints:

$$X_{n^S}^{n^V} \in \{0, 1\}, \forall n^S \in N^S, \forall n^V \in N^V. \tag{10}$$

$$Y_{p^S}^{l^V} \geqslant 0, \forall p^S \in P^S, \forall l^V \in L^V. \tag{11}$$

Constraints (10) and (11) impose the domain constraints on the variables $X_{n^S}^{n^V}$ and $Y_{p^S}^{l^V}$, respectively.

If there exist $X_{n^S}^{n^V}$ and $Y_{p^S}^{l^V}$ that satisfy constraints (2)–(11), then there exists a feasible VNE solution for the VNR; otherwise, no feasible solution exists, and the corresponding healthcare services will be declined.

3.3. Evaluation metrics

In this subsection, four quantitative metrics are defined to evaluate the performance of the proposed method: acceptance ratio (AR), revenue-to-cost ratio (R2CR), substrate node resource utilization rate (NUR), and substrate link resource utilization rate (LUR).

1. The AR is a measure of the proportion of successfully accepted VNRs out of all VNRs received. It can be calculated as follows:

$$\mathrm{AR} = \frac{\mathrm{Num}(VNR_{suc})}{\mathrm{Num}(VNR)}. \tag{12}$$

in which $\mathrm{Num}(VNR_{suc})$ and $\mathrm{Num}(VNR)$ denote the amount of accepted and total VNRs, respectively.

2. The R2CR metric serves to assess the profitability of the VNE method, and its computation is given by the following expressions:

$$\mathrm{R2CR} = \frac{\mathrm{Res_d}(VNR_{suc})}{\mathrm{Res_c}(VNR_{suc})}, \tag{13}$$

$$\mathrm{Res_d}(V) = \sum_{n^V \in N^V} \mathrm{CPU}_{n^V} + \sum_{l^V \in L^V} \mathrm{BW}_{l^V}, \tag{14}$$

$$\mathrm{Res_c}(V) = \sum_{n^S \in N^S} \sum_{n^V \in N^V} X_{n^S}^{n^V} \cdot \mathrm{CPU}_{n^V} + \sum_{l^S \in L^S} \sum_{p^S \in P^S} \sum_{l^V \in L^V} Y_{p^S}^{l^V} \mathbb{I}_{\{l^S \in p^S\}}. \tag{15}$$

in which $\mathrm{Res_d}(VNR_{suc})$ and $\mathrm{Res_c}(VNR_{suc})$ represent the resource demand and consumption of the successfully embedded VN, respectively.

3. The denotations of NUR and LUR are lucid. NUR embodies the mean rate of resource utilization among operable substrate nodes, and LUR embodies the mean rate of resource utilization among operable substrate links.

4. Two-side matching theory overview

Matching plays a vital role in various market operations, such as marriage, recruitment, and school admission [23]. Two-side matching involves two disjoint groups, where members in both groups desire to be paired up with members in the counterpart group. The matching theory aims to satisfactorily pair up the members in both groups [24].

4.1. Two-side matching game

The matching problem in VNE may include two distinct groups: the substrate node set N^S and the virtual node set N^V, which are mutually exclusive ($N^S \cap N^V = \varnothing$). Each member of one group is presumed to possess a comprehensive and transitive preference for all participants in the other group. Specifically, the preference of substrate nodes for virtual nodes and vice versa can be denoted by preference lists P_{S2V} and P_{V2S}, respectively. For instance, the preference list of n_1^S for the participants in N^V can be denoted as $P_{S2V}(n_1^S) = n_2^V \succ_{n_1^S} n_3^V \succ_{n_1^S} \cdots \succ_{n_1^S} n_1^V$. This indicates that n_2^V is the top preference of n_1^S, followed by n_3^V and so forth. The matching game can be expressed as $(N^S, N^V, P_{S2V}, P_{V2S})$, where the objective is to pair up the members of both groups satisfactorily [24].

Definition 1. (One-to-one Matching) A matching μ is a one-to-one correspondence that can be represented as a mapping from $N^S \cup N^V$ onto itself of order two, denoted as $\mu^2(x) = x$, where $\mu(n^S) \in N^V \cup n^S$ and $\mu(n^V) \in N^S \cup n^V$. The mate of x can be represented as $\mu(x)$ [25].

4.2. Stable matching

Marriage is a classic example of a two-sided matching game. In this subsection, we will use the marriage example to introduce the concept of stable matching.

Definition 2. (Block). A block occurs when a player x prefers to remain single rather than be matched with the mate assigned by the matching μ (i.e., $x \succ_x \mu(x)$). Furthermore, if a pair (n^S, n^V) prefer each other over their assigned mates under the matching μ (i.e., $n^S \succ_{n^V} \mu(n^V)$ and $n^V \succ_{n^S} \mu(n^S)$), then the matching μ is blocked by that pair [20].

Definition 3. (Stable Matching). If there are no individuals or pairs that can block the matching μ, then we say that the matching $\mu$is pairwise stable [26].

The literature [14] proposed the Deferred Acceptance algorithm to find a stable marriage matching. In this algorithm, members of one group (men, for example) propose to members of the other group (women, for example) according to their preference list. If a woman receives multiple proposals, she holds the most preferred one and rejects others. Then, the remaining men propose to their most preferred women who have not refused them before, respectively. This cycle repeats until all men are held, or no man is willing to apply. Finally, the algorithm terminates, and each woman is matched with the man she ultimately holds.It is worth noting that there is also a stable matching when women apply to men. In literature [14], the following theorems were proven for stable matching.

Theorem 1. In the marriage market, at least one stable matching always exists.

Theorem 2. When men and women have strict preferences for each other, there always exists a unique man-optimal stable matching and a unique woman-optimal stable matching. In the man-optimal stable matching, each man is matched with his most preferred mate among all stable matchings. Similarly, in the woman-optimal stable matching, each woman is matched with her most preferred mate among all stable matchings. The delayed acceptance algorithm, when applied with men proposing, finds the men-optimal stable matching, and when applied with women proposing, finds the women-optimal stable matching.

5. Matching theory-based VNE method

The challenge of VNE in healthcare network slicing can be likened to the classic marriage matching problem. In this analogy, virtual and substrate nodes play the roles of men and women, respectively. As detailed in subSection 4.1, we employed the virtual and substrate node notations to denote the two sides of the matching game. The resulting matching outcomes reflect the embedding decisions of virtual nodes. For example, should virtual node n_2^V and substrate node n_1^S be paired, n_2^V would opt to embed on n_1^S. However, to obtain such results, we still need to obtain the preference relationship

between substrate and virtual nodes. MT-VNE introduces four metrics to quantify the relationship between virtual nodes and substrate nodes: node resource, location, resource efficiency, and link connectivity preference metrics. The roles of these four metrics in MT-VNE are depicted in Fig. 2. When a VNR arrives, the node and location preference metrics are employed to evaluate the preference of virtual nodes for substrate nodes, while the node resource efficiency and link connectivity metrics are utilized to measure the preference of substrate nodes for virtual nodes. As a result, the preference relationship between the two components can be determined. For example, as illustrated in Fig. 2, virtual node **a** exhibits the highest preference for substrate node **A**, followed by **B** and so on.

5.1. Virtual nodes preference construction

The preference of a virtual node over physical nodes is mainly based on two aspects: the difference in CPU resources between the nodes and the deviation in their locations. Without loss of generality,we consider the virtual node n^V and physical node n^S as an example. The preference value of n^V to n^S can be calculated as follows:

$$Pval_{V2S}(n^V, n^S) = \gamma \cdot R_{V2S}(n^V, n^S) + D_{V2S}(n^V, n^S), \tag{16}$$

in which $R_{V2S}(n^V, n^S)$ and $D_{V2S}(n^V, n^S)$ represent the node resource and location preference index of n^V to n^S. A weight coefficient, γ, is introduced to balance the relative importance of these two indices. Using the computed preference values, n^V will prioritize its preference for n^S in descending order.

Eq. (17) formulates the resource preference index $R_{V2S}(n^V, n^S)$. This index serves as a metric to gauge the relative attractiveness of n^S to n^V, based on the availability of unoccupied CPU resources. Specifically, the resource preference index compares the unoccupied CPU resource of n^S with the demand of n^V, and assigns a value of 1 if the remainder can satisfy the demand, and -1 otherwise. Therefore, n^V exhibits a preference for substrate nodes that possess resources exceeding its demand, in accordance with the resource constraint.

$$R_{V2S}(n^V, n^S) = \begin{cases} 1 & \mathrm{CPU}_{n^S} \geqslant \mathrm{CPU}_{n^V} \\ -1 & \mathrm{CPU}_{n^S} < \mathrm{CPU}_{n^V} \end{cases} \tag{17}$$

The location preference index of virtual nodes can be divided into two parts: the direct location preference index $D^{dir}_{V2S}(n^V, n^S)$ and adjacent location preference index $D^{adj}_{V2S}(n^V, n^S)$. These indexes can be calculated as follows:

$$D_{V2S}(n^V, n^S) = D^{dir}_{V2S}(n^V, n^S) + D^{adj}_{V2S}(n^V, n^S), \tag{18}$$

$$D^{dir}_{V2S}(n^V, n^S) = -\exp(\mathrm{D}(\mathrm{Loc}_{n^V}, \mathrm{Loc}_{n^S})), \tag{19}$$

$$D^{adj}_{V2S}(n^V, n^S) = \sum_{l_{n^V n\prime^V} \in L_V} \max(\{D^{dir}_{V2S}(n\prime^V, n\prime^S) : l_{n^S n\prime^S} \in L_S\}). \tag{20}$$

The direct location preference index can be modeled by (19), which captures the relationship between the distance between virtual and substrate nodes. Notably, value of $D^{dir}_{V2S}(n^V, n^S)$ exhibits a downward trend with increasing $\mathrm{D}(\mathrm{Loc}_{n^V}, \mathrm{Loc}_{n^S})$. Additionally, the exponential function expedites the rate of decay, thereby reinforcing the location constraint and rendering virtual nodes n^V more inclined to select proximal substrate nodes.

The formulation of the adjacent location preference index is provided by (20), wherein $n\prime^V$ and $n\prime^S$ denote the neighboring nodes of n^V and n^S, respectively. The max(·) operator is utilized to identify the maximal element of the set enclosed within the brackets. Thus, $D^{adj}_{V2S}(n^V, n^S)$ endeavors to minimize the summation of distances between the adjacent nodes of n^S and n^V. In essence, the adjacent location preference index heightens the probability of interconnected virtual nodes being embedded onto interconnected substrate nodes, thereby curtailing the hops and bandwidth consumption of substrate paths that virtual links are mapped into.

In conclusion, the substrate nodes that are favored by n^V exhibits three distinct features: (i) an abundance of resources that exceed the demands of n^V; (ii) close proximity to n^V; and (iii) adjacency to nodes that are in close proximity to the adjacent nodes of n^V.

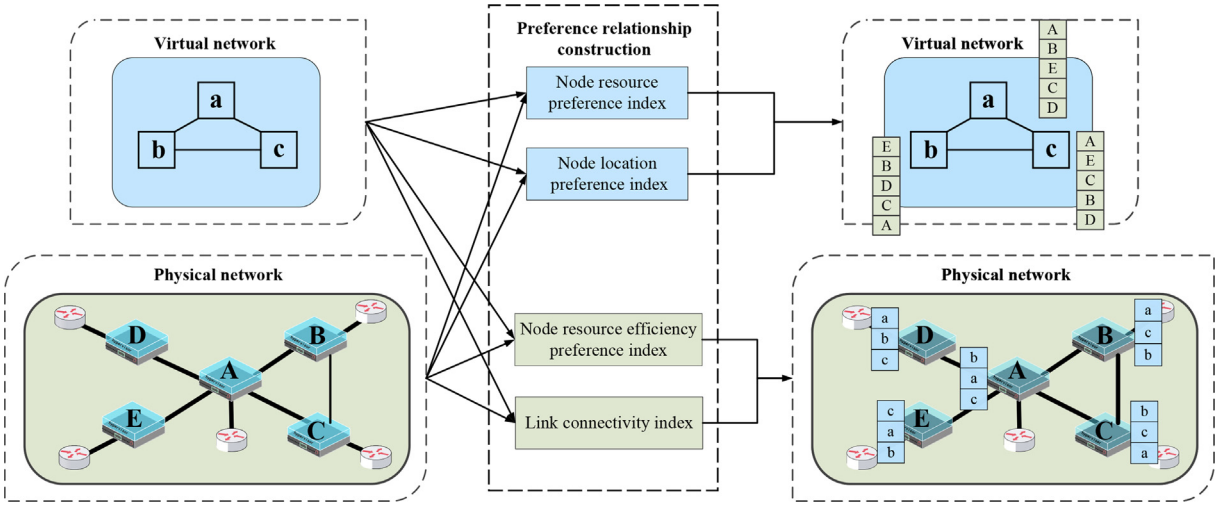

Fig. 2 Four indexes are devised in this work to measure the preference relationship (represented by the lists order) between substrate and virtual nodes.

5.2. Substrate nodes preference construction

The preference of a substrate node over virtual nodes is evaluated based on two factors: node resource utilization efficiency and link connectivity. The preference value of a substrate node n^S to virtual node n^V can be calculated as follows:

$$Pval_{S2V}(n^S, n^V) = \eta \cdot E_{S2V}(n^S, n^V) + L_{S2V}(n^S, n^V). \tag{21}$$

In Eq. (21), $E_{S2V}(n^S, n^V)$ represents the resource utilization efficiency index, $L_{S2V}(n^S, n^V)$ denotes the link connectivity index, and η is a coefficient used to balance the significance of these two factors. These two factors are used to evaluate the preference value of a substrate node over a virtual node.

$$E_{S2V}(n^S, n^V) = \frac{1}{(\mathrm{CPU}_{n^S} - \mathrm{CPU}_{n^V}) + \delta} \tag{22}$$

In Eq. (22), δ is a small positive constant used to avoid division by zero. One may observe that the efficiency index shall produce a negative, small positive, or large positive value contingent upon whether the resource demand of the virtual node exceeds, significantly falls short of, or slightly falls short of the remaining resources of the substrate node. As a consequence of the efficiency index, a substrate node exhibits a preference towards accommodating a virtual node whose resource demand is moderately lower than its remaining resources. This approach helps to make better use of the unoccupied resources in working nodes.

In this investigation, we operate under the assumption that the bandwidth requirements of virtual links cannot be distributed among multiple substrate paths. Consequently, the chosen substrate nodes must possess adjacent links capable of accommodating the bandwidth demands of the corresponding virtual nodes' adjacent links. To satisfy this prerequisite, we introduce the link connectivity index, as depicted in (23), where $C(n^S, l_{n^V n\prime^V})$ determines the number of adjacent links belonging to n^S that can fulfill the bandwidth necessities of $l_{n^V n\prime^V}$. The computation of $C(n^S, l_{n^V n\prime^V})$ is articulated in (24). The link connectivity index is typically a non-negative quantity, which reflects the variety of link embedding alternatives offered by n^S's adjacent links to n^V's adjacent links.

$$L_{S2V}(n^S, n^V) = \sum_{l_{n^V n\prime^V}} C(n^S, l_{n^V n\prime^V}) \cdot \prod_{l_{n^V n\prime^V}} \mathbb{I}_{\{C(n^S, l_{n^V n\prime^V}) \neq 0\}} \tag{23}$$

$$C(n^S, l^V) = \left| \left\{ l_{n^S n\prime^S} \in L^S : \mathrm{BW}_{l_{n^S n\prime^S}} \geqslant \mathrm{BW}_{l^V} \right\} \right| \tag{24}$$

According to the rules established in this subsection, the substrate node n^S will exhibit a preference towards hosting virtual nodes that satisfy the following conditions: (i) possess a resource demand that is marginally lower than the residual resources of n^S, and (ii) have a node embedding that enables a greater number of link embedding alternatives.

Ultimately, the preference lists P_{V2S} and P_{S2V} can be constructed based on the following principles:

$$P_{V2S}(n^V) : n^S \succ_{n^V} n\prime^S \iff Pval_{V2S}(n^V, n^S) > Pval_{V2S}(n^V, n\prime^S) \tag{25}$$

$$P_{S2V}(n^S) : n^V \succ_{n^S} n\prime^V \iff Pval_{S2V}(n^S, n^V) > Pval_{S2V}(n^S, n\prime^V) \tag{26}$$

5.3. The implementation of MT-VNE

The process of online MT-VNE is depicted in Fig. 3. It is worth noting that the arrival and expiration time of VNRs are subject to dynamic changes, rendering them unpredictable. In light of this, the MT-VNE algorithm operates by sequentially mapping or offloading VNRs as they arrive or terminate. At each time slot t, the SN reclaims the resources allocated to the expiring VNRs and offloads the services. MT-VNE is then employed to embed the newly-arrived VNRs. Upon successful mapping, the SN assigns resources to accommodate the VN and updates the occupied and remaining SN resources. The process repeats itself at the next time slot.

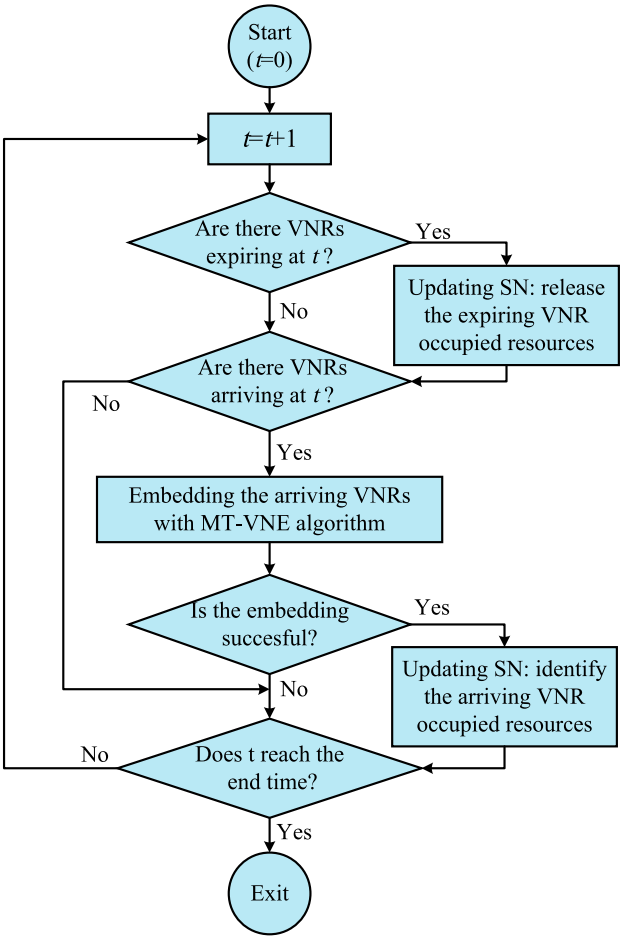

Fig. 3 The diagram of online MT-VNE process.

Algorithm 1. MT-VNE

Input: SN state G^S; arrival VNR G^V
Output: VNE Solution
1: Calculate $Pval_{V2S}$ according to (16)–(20)
2: Calculate $Pval_{S2V}$ according to (21)–(24)
3: Based on the obtained $Pval_{V2S}$ and $Pval_{S2V}$, derive the preference lists P_{V2S} and P_{S2V} using (25) and (26)
4: Execute deferred acceptance algorithm-based virtual node mapping (*Algorithm 2*)
5: **if** The virtual node mapping fails **then**
6: Reject the medical service request
7: **else**
8: Execute modified shortest path-based virtual link mapping (*Algorithm 3*)
9: **if** The virtual link mapping fails **then**
10: Reject the medical service request
11: **else**
12: Accommodate the medical service request
13: **end if**
14: **end if**
15: Output the VNE result

Algorithm 2. Deferred Acceptance Algorithm-based Virtual Node Mapping

Input: The preference lists P_{V2S} and P_{S2V}
Output: The virtual node mapping result
Initialize: Single virtual node set $U^V = N^V$; the match result $\mu = \phi$
1: **while** $U^V \neq \phi$**and** $\exists u^V \in U^V$, s.t.$P_{V2S}(u^V) \neq \phi$ **do**
2: For all $u^V \in U^V$, u^V proposes to its most preferred substrate node according to $P_{V2S}(u^V)$
3: For all $u^V \in U^V$, remove the substrate node that u^V has just applied for in step 2 from $P_{V2S}(u^V)$
4: For each substrate node n^S that receives proposals, it evaluates each proposal and holds the most preferred virtual node among previously held and new applicants, with the guidance of $P_{S2V}(n^S)$
5: Refresh $\mu$with current match pairs
6: **end while**
7: **if** $U^V \neq \phi$ **then**
8: The virtual node mapping fails
9: **else**
10: **if** For all $n^V \in N^V$, the substrate node $\mu(n^V)$ satisfies the location deviation and resource constraints of n^V **then**
11: The virtual node mapping succeeds
12: **else**
13: The virtual node mapping fails
14: **end if**
15: **end if**
16: Output the virtual node mapping result μ

Algorithm 3. Modified Shortest Path-based Virtual Link Mapping

Input: The virtual node mapping result μ
Output: The virtual link mapping result
Initialize: The shortest path set $P^S_{shortest} = \phi$
1: **for** $l_{n^V n\prime^V}$ in L^V **do**
2: Temporarily remove the substrate links that do not meet $l_{n^V n\prime^V}$'s bandwidth constraint
3: Utilizing Dijkstra's algorithm to search for the shortest path $p^S_{\mu(n^V)\mu(n\prime^V)}$ between the substrate nodes $\mu(n^V)$ and $\mu(n\prime^V)$.
4: $P^S_{shortest} = P^S_{shortest} \cup p^S_{\mu(n^V)\mu(n\prime^V)}$
5: **end for**
6: **if** For every virtual link $l^V_{n^V n\prime^V} \in L^V$, the selected substrate path $p^S_{\mu(n^V)\mu(n\prime^V)}$ satisfies both the bandwidth and propagation constraints of $l^V_{n^V n\prime^V}$ **then**
7: The virtual link mapping succeeds
8: **else**
9: The virtual link mapping fails
10: **end if**
11: Output the virtual link mapping result $P^S_{shortest}$

Algorithm 1 outlines the MT-VNE algorithm, which consists of three main steps. First, the preference lists are generated based on the current state of the SN and the arriving VNR (lines 1–3). Then, the deferred acceptance algorithm is executed with the guidance of preference lists to determine the virtual node embedding choices (line 4). If all virtual nodes can be embedded successfully, the modified shortest path-based algorithm is employed to embed the virtual links (line 8).

Algorithm 2 describes the deferred acceptance algorithm-based virtual node mapping, while Algorithm 3 details the modified shortest path-based virtual link mapping. In the virtual node mapping stage, the matching pairs obtained by the deferred acceptance algorithm represent the virtual node embedding choices. If each substrate node in the match pairs can meet its corresponding virtual node's resource and location requirements, the virtual node mapping can be executed successfully. Then, MT-VNE proceeds to the virtual link mapping stage. In the virtual link mapping stage, the modified shortest path-based algorithm first prunes the substrate links that fail to meet the virtual link's bandwidth requirements. Based on the pruned SN, Dijkstra's algorithm is adopted to search for the shortest path between substrate nodes $\mu(n^V)$ and $\mu(n\prime^V)$ to host the virtual link $l_{n^V n\prime^V}$. If all selected substrate paths can fulfill the corresponding virtual links' resource and propagation delay requirements, the virtual link mapping succeeds, and the VNR is embedded successfully. If any mapping fails, the SN declines to provide service to the VNR.

The time complexity of the MT-VNE algorithm can be divided into three parts: (i) calculation of preference lists represented as $\mathcal{O}(|N^S| \cdot |N^V|)$; (ii) deferred acceptance algorithm

Table 2 Substrate network generation parameters.

Network generation method	Waxman random graph, $\alpha = 0.4$ and $\beta = 0.3$
Substrate network size	60 substrate nodes
Node CPU capacity	~U(50,100)
Node location	x and y ~U(0,100)
Link bandwidth capacity	~U(50,100)
Link propagation delay	~U(2,10)

Table 3 VNR generation parameters.

Network generation method	Waxman random graph, $\alpha = 0.4$ and $\beta = 0.3$
VNRs' arriving rate	Poisson process, with a mean value of 50 VNRs per 100 time units
VNRs' life time	Exponential distribution, with a mean value 200 time units
VNRs' node numbers	~DiscreteU(2,10)
Node CPU demand	~U(1,20)
Node preferred location	x and y ~U(0,100)
Node distance constraint	~U(10,20)
Link bandwidth demand	~U(1,20)
Link propagation delay constraint	Experiment 1: no delay constraint Experiment 2: ~U(10,30)

based virtual node mapping as $\mathcal{O}(|N^S| \cdot |N^V|)$; (iii) modified shortest path-based virtual link mapping as $\mathcal{O}(|L^S| \cdot |L^V| \cdot \log |N^S|)$. Overall, MT-VNE can run in polynomial time.

6. Performance evaluation

In this section, we evaluate the performance of MT-VNE and compare it with three baselines to show its efficiency in medical service provisioning.

6.1. Simulation settings

Table 2 and 3 present the parameters utilized in generating the underlying 5G SN and VNRs representing healthcare services for the simulation experiments. Most of the parameters used in our simulation experiments are comparable to those used in previous studies [5,15,27]. Notably, we integrated delay and location requirements to tackle the distinctive aspects of medical network slicing scenarios.

The coefficients used in (13), (18), and (19) are set based on pre-commissioning, with $\gamma = 500, \eta = 200$, and $\delta = 0.0001$. All simulations were conducted on a server equipped with two Intel Core Xeon E5-267@2.5 GHz processors and 128G RAM.

6.2. Simulation result

In this subsection, we compare the simulation results of different algorithms: NTANRC [5], CPSO-VNE [6], TD-VNE [15], and the proposed MT-VNE. We present the evaluation metrics, including AR, R2CR, NUR, and LUR, of these four methods to compare their performance.

Fig. 4 depicts the performance metric variation curves of diverse algorithms, acquired from experiment 1 (devoid of delay constraint) and experiment 2 (with delay constraint), in an order that has been adjusted to promote comparison between the two experiments. Both experiments incorporate the location constraint. The results evince that MT-VNE surpasses other algorithms in terms of acceptance ratio (AR), node resource utilization rate (NUR), and link resource utilization rate (LUR). This is owing to the fact that conventional VNE algorithms do not consider location constraints, rendering them unsuitable for healthcare service provisioning. Furthermore, the location preference index in MT-VNE facilitates the mapping of virtual nodes to substrate nodes within the required location range of services, thereby enhancing the acceptance rate and resource utilization rate.

After analyzing the simulation results of experiment 1 shown in Fig. 4(a)-(d), we can draw the following conclusions: (i) The MT-VNE algorithm outperforms the other three algorithms in terms of AR, NUR, and LUR metrics. In addition, the MT-VNE's AR metric in Fig. 4(a) approaches 1, indicating high embedding success rate and efficient substrate resource utilization. (ii) The CPSO-VNE algorithm achieves the highest R2CR, which is reasonable as its objective is to find the VNE solution with the lowest cost. However, CPSO-VNE's highest R2CR value does not necessarily lead to accommodating more VNRs, which is evident from its second-lowest AR metric in Fig. 4(a). In contrast, MT-VNE achieves the second-highest R2CR value and outperforms CPSO-VNE in the AR value by a considerable margin. (iii) The metric values obtained by MT-VNE significantly outperform the node ranking based-method NTANRC in Fig. 4(a)-(d), which confirms the effectiveness of introducing matching theory into healthcare-oriented VNE problems.

By analyzing the simulation results of experiment 2 presented in Fig. 4(a)-(d), and comparing them with Fig. 4(a)-(d), we can draw the following conclusions: (i) The appearance of delay constraint in experiment 2 leads to inferior AR, NUR, and LUR metric values compared to experiment 1, while the R2CR metric values show improvement. The increase in VNE difficulty due to the delay constraint leads to a drop in AR, NUR, and LUR metric values, while the delay constraint restricts the virtual link embedding cost, thus increasing the R2CR metric value. (ii) In experiment 2, the MT-VNE algorithm continues to outperform the other algorithms on AR, NUR, and LUR metrics. Moreover, the gap between the R2CR values of MT-VNE and CPSO-VNE is further narrowed. (iii) Comparing Fig. 4(a)-(d), and Fig. 4(e), we can conclude that the drop in AR metric values of CPSO-VNE is the least when the delay constraint is added, as demonstrated by the reduced gap between the AR metric values of CPSO-VNE and NTABRC. This is because the objective function of CPSO-VNE is to minimize the virtual link embedding cost, which is in line with the delay constraint to some extent.

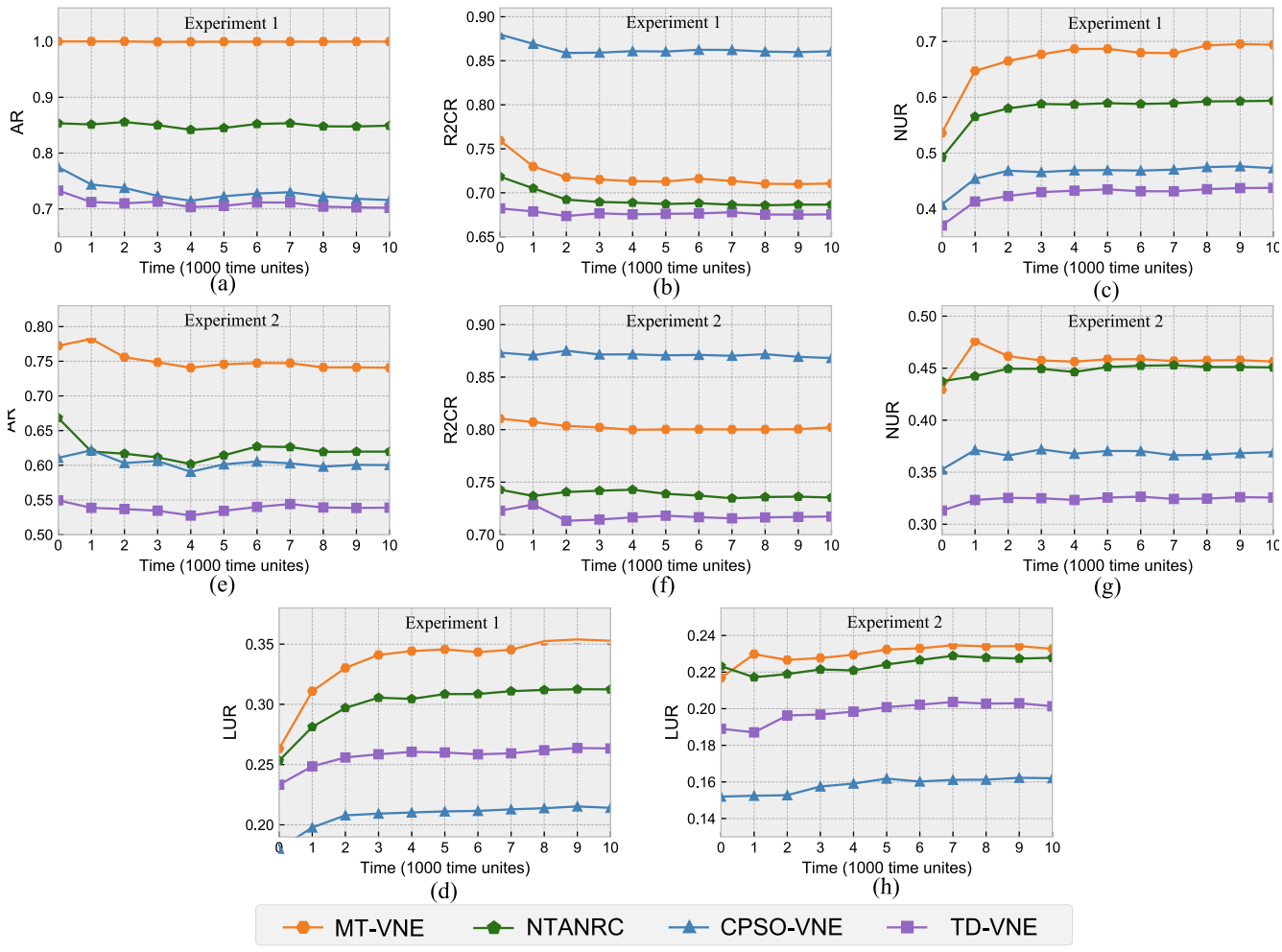

Fig. 4 The performance metrics garnered from experiment 1 and experiment 2 for various algorithms are illustrated in Figs. 4(a)-(d) and 4(e)-(h), respectively. Please note that (i) the order of sub-figures has been modified for the sake of facilitating a comparison between experiment 1 and experiment 2, and (ii) all sub-figures share a common legend located at the bottom of the figure..

Table 4 Average running time of different algorithms.

	CPSO-VNE	NTANRC	TD-VNE	MT-VNE
Average map time (Experiment 1)	6.459s	0.445s	0.440s	0.084s
Average map time (Experiment 2)	3.897s	0.451s	0.398s	0.095s

In addition to evaluating the VNE performance metrics, we also analyze the average execution time of different algorithms, as presented in Table 4. The results show that CPSO-VNE algorithm spends the most time in embedding a service, which is expected as it is a meta-heuristic method. TD-VNE algorithm consumes slightly less time than NTANRC. Notably, MT-VNE algorithm takes the least time to embed a VNR, accounting for only 20.0% of NTANRC, 21.3% of TD-VNE, and 1.73% of CPSO-VNE. In summary, MT-VNE algorithm not only achieves excellent performance in VNE evaluation metrics but also demonstrates significantly faster execution time than the other algorithms.

6.3. Preference index utility analysis

In addition to the performance evaluation experiments, we designed an experiment to investigate the impact of the four preference indexes on MT-VNE. This experiment included five simulations, namely, the original MT-VNE and four modified versions of MT-VNE, each with one preference index removed. All simulations were executed with the propagation delay constraint, and the results are shown in Fig. 5.

In Fig. 5, R_{V2S} and D_{V2S} represent the node resource and location preference indexes of the virtual node to the substrate node, respectively, while E_{S2V} and L_{S2V} denote the node

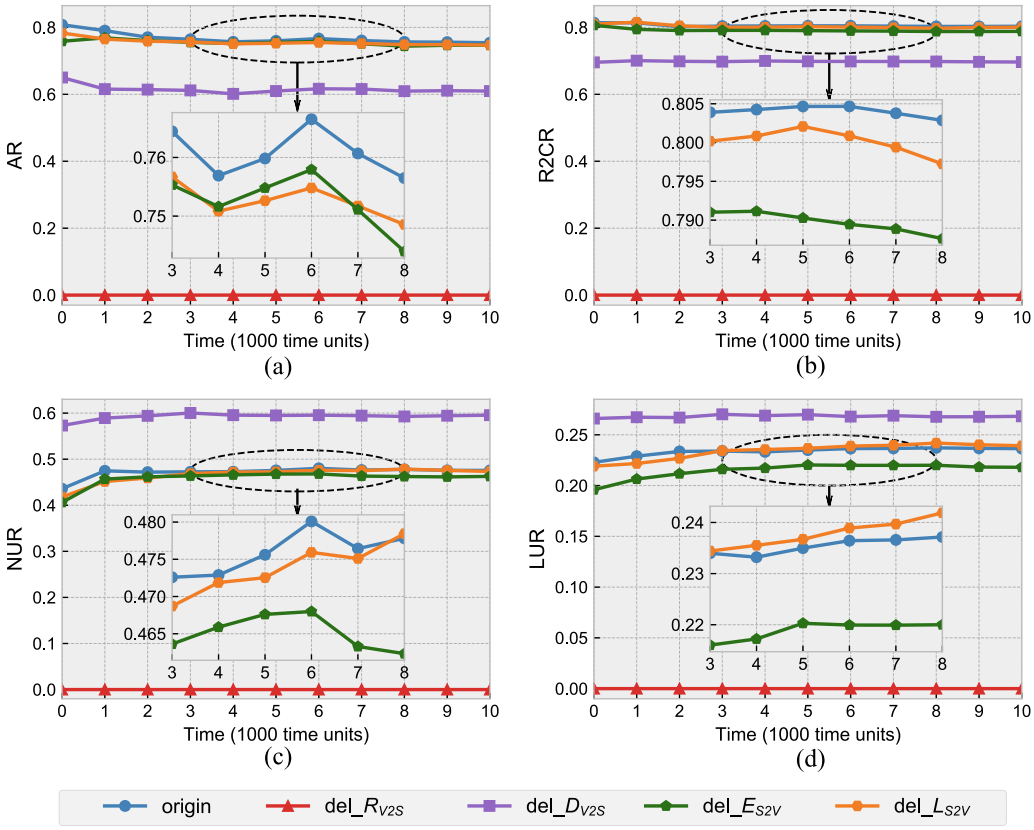

Fig. 5 Simulation results depict the performance of the unaltered MT-VNE algorithm and four modified versions with the omission of one of four preference indices. Please note that (i) del_R_{V2S}, del_D_{V2S}, del_ E_{S2V}, and del_L_{S2V} represent the modified algorithms with corresponding indexes removed and (ii) all sub-figures share a common legend located at the bottom of the figure..

resource utilization efficiency and link connectivity preference indexes of the substrate node to the virtual node, respectively. By comparing the simulation results in Fig. 5, we can conclude the effects of each preference index as follows. (i) As previously mentioned in Section 5-A, -A, R_{V2S} is crucial in ensuring the satisfaction of the node resource constraint. In Fig. 5, all four metrics simultaneously drop to zero when deleting R_{V2S}, demonstrating its fundamental role in MT-VNE. (ii) Removing D_{V2S} leads to a significant reduction in AR and R2CR. This is reasonable since D_{V2S} helps MT-VNE in reducing the hops and bandwidth consumption of substrate paths that virtual links are embedded in, which can improve the VNRs' acceptance ratio as well as the revenue to cost ratio. (iii) E_{S2V} has a positive impact on all four evaluation metrics, while L_{S2V} only benefits AR, R2CR, and NUR metric values. However, compared with the preference indexes of virtual node to substrate node (R_{V2S} and D_{V2S}), the preference indexes of substrate node to virtual node (E_{S2V} and L_{S2V}) have a relatively smaller effect on MT-VNE, as demonstrated by their closer results to the original MT-VNE.

7. Conclusion

This study aims to address the challenge of healthcare service provisioning by incorporating network slicing architecture and modeling it as a virtual network embedding (VNE) problem. Unlike traditional VNE problems, the medical network slicing-oriented VNE problem is distinguished by its distinctive features, which include the imposition of delay and location constraints. In light of this, we proposed an algorithm that is based on matching theory, known as the Matching Theory-based VNE (MT-VNE) algorithm, in this work. In MT-VNE, four preference indexes that take into account VNE problem objectives and constraints are devised to construct preference lists. The deferred acceptance algorithm is then utilized to obtain virtual node mapping results, followed by virtual link mapping with the modified shortest path algorithm. Experimental results demonstrate that MT-VNE outperforms baselines in terms of medical service acceptance ratio and physical resource utilization, especially when considering the delay constraint. Moreover, MT-VNE shows significant reductions in VNE execution time. Future research will investigate the effects of substrate and virtual network topology characteristics and load variations of healthcare services on the proposed approach.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the King Saud University, Riyadh, Saudi Arabia, through the Researchers Supporting Project under Grant RSP2023R18.

References

[1] A. Acemoglu, J. Krieglstein, D.G. Caldwell, F. Mora, L. Guastini, et al, 5g robotic telesurgery: Remote transoral laser microsurgeries on a cadaver, IEEE Trans. Med. Robot. Bion. 2 (4) (2020) 511–518.

[2] A. Moglia, K. Georgiou, B. Marinov, E.s. Georgiou, R.N. Berchiolli, R.M. Satava, A. Cuschieri, 5g in healthcare: From covid-19 to future challenges, IEEE J. Biomed. Health Inform. 26 (8) (2022) 4187–4196.

[3] Y. Djenouri, A. Belhadi, G. Srivastava, Jerry C. Lin, Secure collaborative augmented reality framework for biomedical informatics, IEEE Journal of Biomedical and Health Informatics 26 (6) (2022) 2417–2424.

[4] A. Song, W. Chen, Y. Gong, X. Luo, J. Zhang, A divide-and-conquer evolutionary algorithm for large-scale virtual network embedding, IEEE Trans. Evol. Comput. 24 (3) (2020) 566–580.

[5] H. Cao, L. Yang, H. Zhu, Novel node-ranking approach and multiple topology attributes-based embedding algorithm for single-domain virtual network embedding, IEEE Internet of Things Journal 5 (1) (2018) 108–120.

[6] A. Song, W.N. Chen, T. Gu, H. Zhang, J. Zhang, A constructive particle swarm optimizer for virtual network embedding, IEEE Transactions on Network Science and Engineering 7 (3) (2020) 1406–1420.

[7] Khoa Nguyen, Changcheng Huang, Toward Adaptive Joint Node and Link Mapping Algorithms for Embedding Virtual Networks: A Conciliation Strategy, IEEE Trans. Netw. Serv. Manage. 19 (3) (2022) 3323–3340.

[8] P. Zhang, X. Pang, G. Kibalya, N. Kumar, S. He, B. Zhao, Gcmd: Genetic correlation multi-domain virtual network embedding algorithm, IEEE access 9 (2021) 67167–67175.

[9] Shubin Zhang, Hui Gu, Kaikai Chi, Liang Huang, Keping Yu, Shahid Mumtaz, DRL-based partial offloading for maximizing sum computation rate of wireless powered mobile edge computing network, IEEE Trans. Wireless Commun. 21 (12) (2022) 10934–10948.

[10] P. Zhang, C. Wang, C. Jiang, A. Benslimane, Security-aware virtual network embedding algorithm based on reinforcement learning, IEEE Transactions on Network Science and Engineering 8 (2) (2021) 1095–1105.

[11] Z. Yan, J. Ge, Y. Wu, L. Li, T. Li, Automatic virtual network embedding: A deep reinforcement learning approach with graph convolutional networks, IEEE J. Sel. Areas Commun. 38 (6) (2020) 1040–1057.

[12] Peiying Zhang, Peng Gan, Neeraj Kumar, Ching-Hsien Hsu, Shigen Shen, Shibao Li, RKD-VNE: Virtual network embedding algorithm assisted by resource knowledge description and deep reinforcement learning in IIoT scenario, Future Generation Computer Systems 135 (2022) 426–437.

[13] Yinzhi Lu, Liu Yang, Simon X Yang, Qiaozhi Hua, Arun Kumar Sangaiah, Tan Guo, Keping Yu, An Intelligent Deterministic Scheduling Method for Ultralow Latency Communication in Edge Enabled Industrial Internet of Things, IEEE Trans. Industr. Inf. 19 (2) (2022) 1756–1767.

[14] D. Gale, L.S. Shapley, College admissions and the stability of marriage, The American Mathematical Monthly 69 (1) (1962) 9–15.

[15] S. Wang, J. Bi, J. Wu, A.V. Vasilakos, Q. Fan, Vne-td: A virtual network embedding algorithm based on temporal-difference learning, Comput. Netw. 161 (2019) 251–263.

[16] A. Vergutz, G. Noubir, M. Nogueira, Reliability for smart healthcare: A network slicing perspective, IEEE Network 34 (4) (2020) 91–97.

[17] P.I. Tebe, G. Wen, J. Li, Y. Yang, W. Tian, J. Chong, W. Zhang, 5g-enabled medical data transmission in mobile hospital systems, IEEE Internet of Things Journal 9 (15) (2022) 13679–13693.

[18] D. Lin, Y. Tang, Edge computing-based mobile health system: Network architecture and resource allocation, IEEE Syst. J. 14 (2) (2020) 1716–1727.

[19] P. Zhang, H. Yao, Y. Liu, Virtual network embedding based on computing, network, and storage resource constraints, IEEE Internet Things J. 5 (5) (2018) 3298–3304.

[20] A. Fischer, J.F. Botero, M.T. Beck, H. de Meer, X. Hesselbach, Virtual network embedding: A survey, IEEE Commun. Surv. Tutor. 15 (4) (2013) 1888–1906.

[21] H. Cao, J. Du, H. Zhao, D.X. Luo, N. Kumar, L. Yang, F.R. Yu, Resource-ability assisted service function chain embedding and scheduling for 6g networks with virtualization, IEEE Trans. Veh. Technol. 70 (4) (2021) 3846–3859.

[22] F. Malandrino, C.F. Chiasserini, C. Casetti, G. Landi, M. Capitani, An optimization-enhanced mano for energy-efficient 5g networks, IEEE/ACM Trans. Networking 27 (4) (2019) 1756–1769.

[23] A. Roth, Deferred acceptance algorithms: History, theory, practice, and open questions, Int. J. Game Theory 36 (2008) 537–569.

[24] J. Ding, J. Cai, Two-side coalitional matching approach for joint mimo-noma clustering and bs selection in multi-cell mimo-noma systems, IEEE Trans. Wireless Commun. 19 (3) (2019) 2006–2021.

[25] A.E. Roth, M.A.O. Sotomayor, Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis. Econometric Society Monographs, Cambridge University Press, Cambridge, 1990.

[26] M. Touati, R. El-Azouzi, M. Coupechoux, E. Altman, J. Kelif, A controlled matching game for wlans, IEEE J. Sel. Areas Commun. 35 (3) (2017) 707–720.

[27] P. Zhang, X. Pang, Y. Bi, H. Yao, H. Pan, N. Kumar, Dscd: Delay sensitive cross-domain virtual network embedding algorithm, IEEE Trans. Network Sci. Eng. 7 (4) (2020) 2913–2925.